Multidomain Design and Optimization based on Comsol Multiphysics: Applications for Mechatronic Devices

A. Bissal[1*], O. Craciun[2], V. Biagini[2], J. Magnusson[3]
[1]ABB AB Corporate Research, Västerås, Sweden
[2]ABB AG Corporate Research, Ladenburg, Germany
[3]Royal Institute of Technology (KTH) Stockholm, Sweden

*Corresponding author: 11423 Stockholm, Sweden, ara.bissal@se.abb.com

Abstract: Although the Finite Element Method (FEM) has been proven to be a powerful tool, it still has limitations. FEM is computationally heavy and can be at times unstable especially when solving highly nonlinear coupled problems. Thus it is very challenging and time consuming to run parametric studies. Consequently, a novel smart and parallelizable algorithm (SPA) was developed. This algorithm can detect failed simulations and take countermeasures as well as perform parallel computations. To test the algorithm, SPA was coupled to COMSOL Multiphysics and was able to successfully solve 20,520 simulations in seven weeks without crashing. If solved linearly, around six months would have been required. COMSOL Multiphysics has proven to be a powerful since it is able to accurately model phenomena using different physics. Coupling SPA with COMSOL opens up new possibilities enabling smart and parallelizable FEM simulations for fast prototyping.

Keywords: Smart, parallel computing, actuator, electromagnetics, mechanics, multi-physics, capacitor, armature, magnet

1. Introduction

With all the recent advances in computers, Finite Element Method based simulations are becoming a standard prior to designing and prototyping. Not only can people decrease the cost of building new prototypes, the entire process from designing to manufacturing has been speeded up due to FEM simulations.

Although FEM is very accurate and can be used to design novel prototypes quickly, it still has several limitations. COMSOL Multiphysics has unique capabilities enabling the modeling of new prototypes combing different physics. Although this increases accuracy and provides the capability to visualize and pinpoint co-dependent phenomena from different physics, it is still a challenge to guarantee convergence when performing parametric studies. Small parametric changes in the model can lead to singularities or longer computational times simply because the model will have a hard time to converge. One such example is when varying the frequency or the rise time of a power source. Increasing the input frequency will certainly require a mesh refinement to accurately capture faster transients and resolve smaller skin depths. A rule of thumb exists for how small the mesh size should be. If the exciting source is frequency based, then the skin depth can be approximated by:

$$\delta_s = \sqrt{\frac{2}{\omega\mu\sigma}} \tag{1.1}$$

where, δ_s is the skin depth, ω is the angular frequency ($2\pi f$), μ is the magnetic permeability, and σ is the electrical conductivity of the medium. It is a good practice to have three mesh elements to resolve the calculated skin depth. Another example could be when adding nonlinear equations to study the influence of some components on system performance. To ensure convergence, a smaller time step might be needed along with a small relative and/or absolute error tolerance.

Such complications arising from parametric variations, poses significant challenges to automate the task and carry out a large number of parametric studies. The risk that the model crashes when automated is quite high. One common way to be able to conduct such large parametric studies is to have the user implement some changes manually and test whether the model converges, solves, and gives reasonable results. This method however consumes a lot of time, valuable time that can be invested in performing something else. Another major limitation is that it is hard to conduct a large number of parametric variations since it would consume a great amount of time and effort. Thus, optimizing the product at hand becomes really difficult.

Before purchasing equipment, it is crucial to carry out some parametric studies to build a good design from the beginning. One such example could be designing a capacitor bank for energizing different types of electromagnetic actuators such as the Thomson coil (TC) or magnetic actuators. Capacitor banks are quite expensive and bulky. Moreover, the performance of these actuators is quite sensitive to the capacitance and charging voltage. Consequently, exploring a large range of parameters can help identify possible contours encompassing the desired optimal values.

This paper presents a novel smart parallelizable algorithm that overcomes these presented challenges, and makes use of parallel computation to speed up the design process.

2. Use of COMSOL Multiphysics

COMSOL Multiphysics has been used to model two electromagnetic actuators, a Thomson coil, and a magnetic actuator.

The TC consists of a multi-turn spirally shaped flat coil with a conductive armature located in close proximity, preferably, directly on top or on the bottom of the coil [1]. A capacitor bank is used to power such an actuator since a large impulse current is required. Currents in the order of several tens of kilo amperes with a rise time of few hundreds of microseconds are crucial to generate a time varying magnetic field. The time derivative of the axial component of the magnetic flux density induces azimuthal currents in the armature. The product of these currents along with the radial component of the magnetic flux density generates a repulsive force, commonly known as a Lorrentz force, oriented in the axial direction. These forces, that are in the range of tens to hundreds of kilonewtons drives the armature and ensures a displacement of tens of millimeters in a couple of milliseconds. Such kind of actuators have received a lot of attention especially for high voltage direct current (HVDC) breakers where a prompt contact separation is crucial to ensure fault currents are interrupted before they attain dangerously large values [2].

The magnetic actuator, used in medium voltage switches, consists of a coil, a small iron disk, a larger iron disk, a pair of magnets, opening springs, and a yoke surrounding the magnet and the coil to decrease system reluctance. Two modes of operation are desired, a closing operation whereby the large disk is attracted towards the yoke, and an opening operation, whereby the large disk is repelled away from the yoke with the help of opening springs [3]. For maintaining a closed position, the holding force from the magnets attracts the small disk and counteracts the opening springs. When the switch is required to open, a current pulse is discharged to counteract the magnet field from the magnets such that the opening springs can drive the large iron disk away from the yoke. Thus, when the yoke reaches its end destination, the coil is no longer powered. Thus it maintains an open position. For the closing operation, a current is discharged in the coil in the opposite direction such that its magnetic field is in line with the magnet's magnetic field. Both these fields superimpose and consequently attract the big disk towards the yoke. This is called a closing operation. The larger the current pulse, the faster is the closing operation. As for the opening operation, special attention has to be made to limit the peak current such that the permanent magnets are not demagnetized.

To model the TC actuator, Multiphysics based simulations are crucial. To model the TC, a two dimensional axis-symmetric model with the following COMSOL modules where used: Solid Mechanics, Magnetic Fields, Moving Mesh, Electrical Circuit, Global ODEs and DAEs, Heat Transfer in Solids, and Weak Form PDE. The computed temperature was used to change the electrical conductivity of the material according to the following equation:

$$\sigma = \sigma_0(1+\alpha(\mathrm{T}-\mathrm{T}_0))^{-1} \qquad (1.2)$$

where, σ_0 is the electrical conductivity at temperature T_0, the room temperature. The temperature coefficient is given by α.

To model the magnetic actuator, the following COMSOL modules were used: Magnetic Fields, Moving Mesh, Electrical Circuit, Global ODEs and DAEs.

3. Methodology

To able to conduct a large number of simulations to explore the space of desirable solutions, a smart and parallelizable algorithm has been developed. The algorithm detects the number of physical cores available on a system. It then starts a MATLAB script "the parent script" and asks for a list of inputs and number of dimensions to create a full factorial. In the example given in this paper, three dimensions where studied at the same time. The TC's number of turns, the capacitor bank's capacitance, and charging voltage were varied. The capacitance was varied from 1 mF to 30 mF in steps of 1 mF. The charging voltage was varied from 200 V to 2000 V in steps of 100 V. The coil's number of turns was varied from 5 turns to 40 turns in steps of 1 turn. This amounts to 30 values for the capacitance, 19 for the voltage, and 36 for the number of turns. The full factorial for this problem amounts to 20,520 multi-physics FEM simulations. The important output variables that are of interest are the final velocity of the armature, the input electrical energy, and the actuator's efficiency.

To perform all 20,520 simulations, a small cluster was used. The cluster consisted of an Intel Xeon processor E5-1620 with 4 physical cores and 8 logical processors. The CPU clock frequency was 3.6 GHz and the total amount of installed random access memory was 32 GB.

After collecting the inputs, the algorithm asks for a MATLAB file of the COMSOL model. It then creates a file filled with tokens each representing each simulation. Afterwards, it creates children equal to the number of available physical processors available on the computer. Each child starts a MATLAB in the background with no graphical user interface and starts a COMSOL server on one of the cores to be able to establish a link between MATLAB and COMSOL. After all the children scripts are initiated, the parent script dies to free up memory. All children are able to communicate with each other. If a child is jobless, it requests for a token. If it gets a unique key identifier and is the only one to have the key, then it may access the file filled with tokens. It is not enough to check if a child has a key to grant access. There might be a case were several children requested for the unique key at the same time. Then another check is done. If two or more children have received a key, then they all lose their keys and wait for a time equal to the child number multiplied by 10 ms. Thus each child will have a unique time delay. This process is iterated by all children until the list of tokens created by the parent is exhausted.

Another functionality of this algorithm, the reason it is called smart, is because it can identify failed simulations and take evasive action. In this first version of the algorithm, the software is able to identify failed cases and tighten the relative and absolute tolerances in stages for these cases. The algorithm then retries to solve the failed case with a tighter tolerance. Setting a tighter tolerance forces COMSOL to solve the model with a smaller time step. Although this increases computation time, it helps for convergence. Most often, the cases that fail to convergence are cases that exhibit extreme nonlinear behavior thus making it hard for the COMSOL solver to converge. In such a case, one of the children might have to retry to solve one particular case several times needing a considerably longer time to finish its work package in comparison with its siblings. However, this is not a problem at all since its siblings will not wait for it. As soon as they finish their work package, they proceed and take another work package from the list of tokens.

4. Simulations and Experimental Validation

Figure 1 shows the velocity of a TC 150 µs following the discharge of a capacitor bank. It can be seen that most of the forces are created in the mushroom directly on top of the coil. Thus these domains attain maximum velocity causing the mushroom armature to bend prior any movement. Although these domains attain speeds up to 38 m/s, the bottom of the armature is almost stationary having a velocity of 0 m/s.

Figure 2 shows a comparison between measurements and simulations for a TC following the discharge of a capacitor bank charged with different voltage levels. The prototype was tested with charging voltages starting from 300 V increasing to 900 V in steps of 100 V. As the charging voltage is increased, the steady state velocity of the TC also increase since a larger current pulse is created.

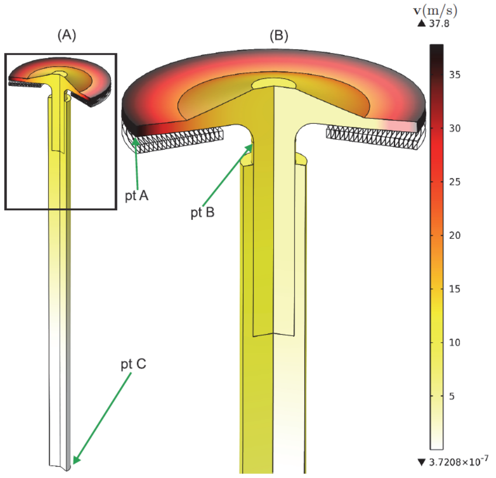

Figure 1. A Picture showing the velocity profile in m/s of the armature of the TC 150 μs after discharge.

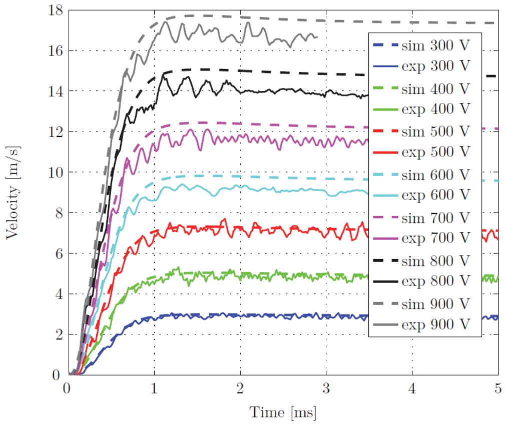

Figure 2. A comparison between simulations and measurements for different charging voltages.

Figure 3 shows the magnetic flux density in the magnetic actuator. A 3D simulation was carried out to model the open and close operations of this actuator. This picture shows a closing operation where the large disk has crossed more than half of its total stroke. Following the discharge of a current pulse in the coil, a magnetic field is created that circulates in the yoke and the air gap separating the yoke from the big disk. This creates an attractive force attracting the large disk towards the yoke. Since a large magnetic field is created, the small disk (not shown in the plot) saturates causing the attractive forces to the large disk to be dominating.

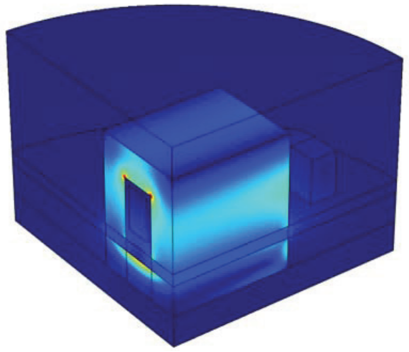

Figure 3. A 3D transient simulation showing the magnetic flux density distribution in (T), for the magnetic actuator. From a top to bottom perspective, the yoke, the coil, a variable air gap (segmented into three sections), and the large disk.

5. Results

This section presents the main results of this paper from the Smart Parallelizable Algorithm algorithm. The total simulation time to conduct all 20520 simulations was 7 weeks. The algorithm was able to solve all cases successfully. Some were solved from the first attempt, some from the second attempt, and some from the third attempt. Not one of the 20,520 simulations failed.

The advantage of such an algorithm is the tremendous speed up in number of simulations. The same algorithm was also implemented and tested on a machine with Linux Open Suse with 16 cores. All 16 cores were able to communicate with each other and successfully solve all jobs. After testing the algorithm, the speed up was almost proportional to the number of available cores. Another advantage is that the children don't wait for each other and continuously work until the list of tokens is exhausted.

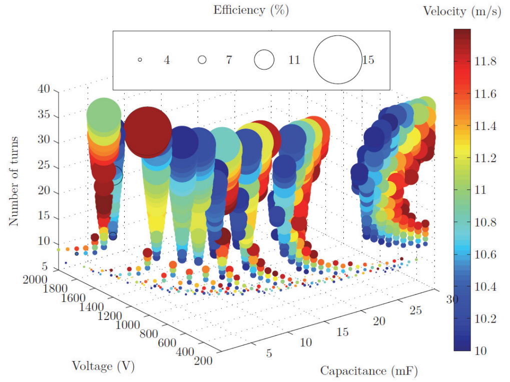

Figure 4. A 5D plot showing the capacitance on the x-axis in mF, the charging voltage on the y-axis in V, the number of turns on the z-axis, the end velocity in m/s in color, and the efficiency in %, in bubble size.

Some sample outputs from the algorithm can be seen in Figure 4 and Figure 5. Figure 4 shows a 5D plot showing the capacitance on the x-axis in mF, the charging voltage on the y-axis in V, the number of turns on the z-axis, the end velocity in m/s in color, and the efficiency in %, in bubble size. Only a portion of the output was chosen to be shown. Thus in this case, a velocity between 10 m/s and 12 m/s was required. Evidently, the optimum is the biggest red bubble that has the largest end velocity and the highest efficiency.

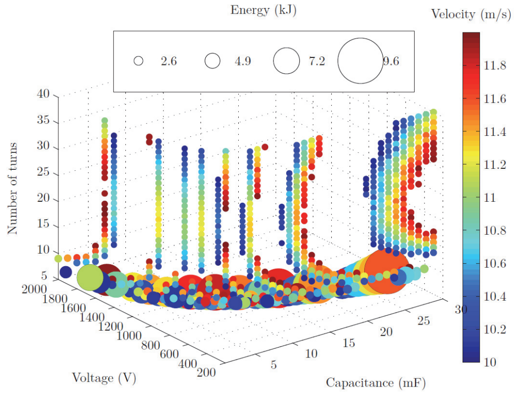

Figure 5. A 5D plot showing the capacitance on the x-axis in mF, the charging voltage on the y-axis in V, the number of turns on the z-axis, the end velocity in m/s in color, and the input energy in kJ, in bubble size.

Similarly, Figure 5 shows a 5D plot showing the capacitance on the x-axis in mF, the charging voltage on the y-axis in V, the number of turns on the z-axis, the end velocity in m/s in color, and the Energy in kJ, in bubble size. This is yet another interesting case where the objective is to minimize the input energy while making sure a velocity between 10 m/s to 12 m/s is attained. Evidently, the optima are the small bubbles with a dark red color. It can be inferred from this graph that using five turns requires a large input energy. To reduce the required input energy, the number of turns should be increased.

6. Conclusions and Future Work

This algorithm was proven to be very stable and was able to provide a speed increase almost proportional to the number of available cores. Moreover, due to its smart functionality, it is very stable and was able to solve all cases successfully. This shows that a smart parallelizable algorithm can be developed.

The presented algorithm is still at its infancy and has a great potential to be expanded. In the future, this brute force algorithm will be combined with a dedicated multi-objective algorithm. Integrating a multi-objective optimization means that the algorithm will be selective and will only solve for cases close to the identified optima. Coupling such a tool with COMSOL Multiphysics, a powerful multi-physics FEM software opens up a new frontier enabling fast prototyping of novel ideas.

7. References

1. A. Bissal, J. Magnusson, G. Engdahl, Comparison of Two Ultra-Fast Actuator Concepts, *IEEE Transactions in Magnetics*, **Volume 48**, 3315 – 3318 (2012)

2. C. M. Frank, HVDC circuit breakers: A review identifying future research needs, *IEEE Transactions on Power Delivery*, **Volume 26**, (2011)

3. O. Craciun, V. Biagini, Multi-Domain Simulation and Analysis of Electromagnetically Actuated Reclosers, *39th Annual Conference of the IEEE Industrial Electronics Society, IECON 2013*, Page(s): 4222 - 4227